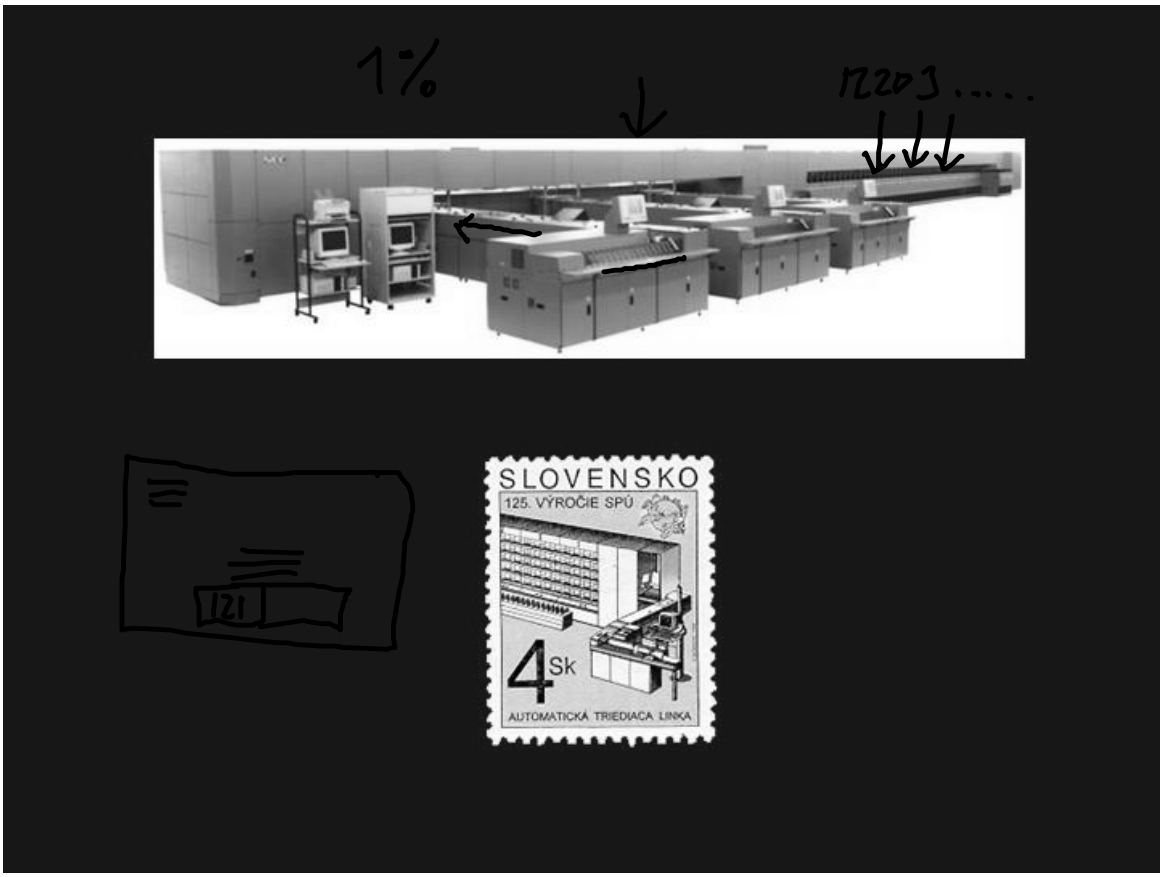
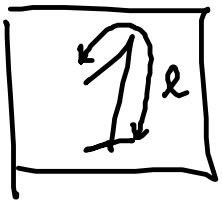
Klassifizierung

0 1 2 3 4 5 6 7 8 9

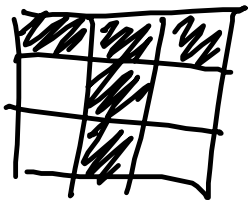
0 1 2 3 4 5 6 7 8 9



Nearest Neighbor

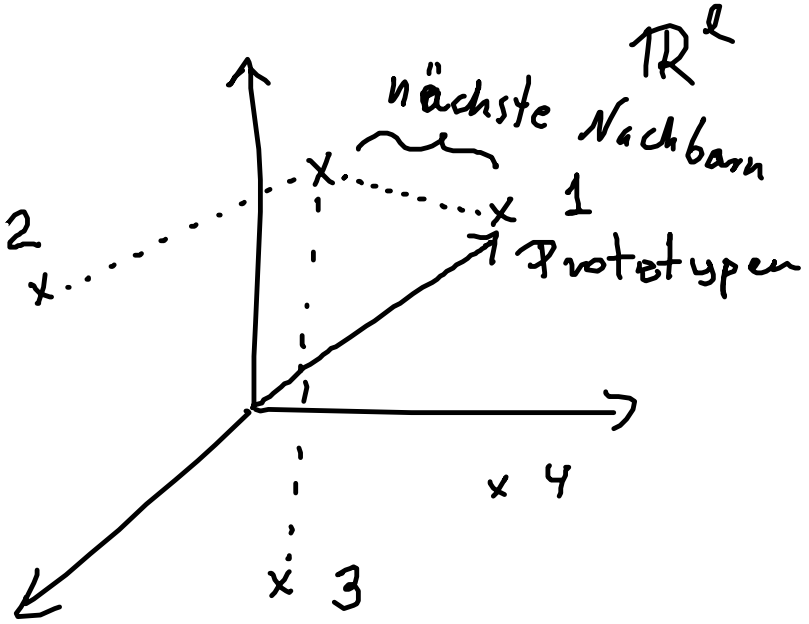


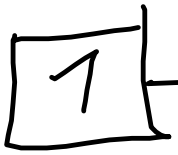
→ Parameter $\begin{cases} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_l \end{cases}$ l



→ 1 1 1 0 1 0 0 1 0

Parameterraum

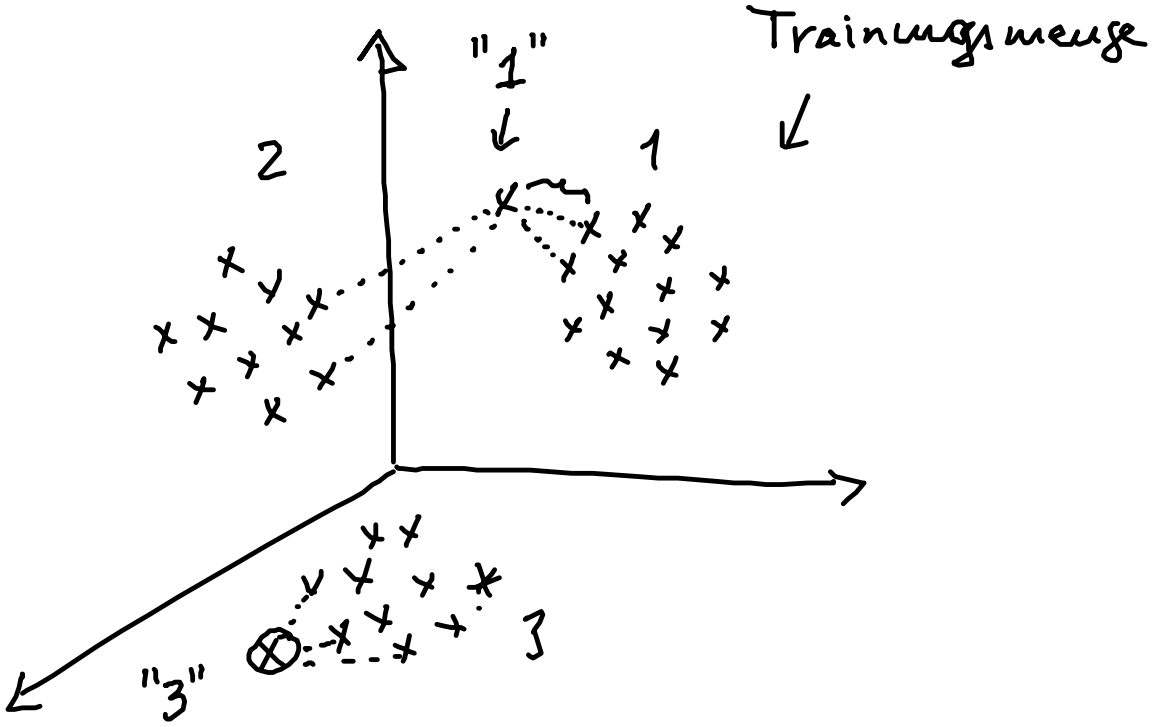




→ Parameter

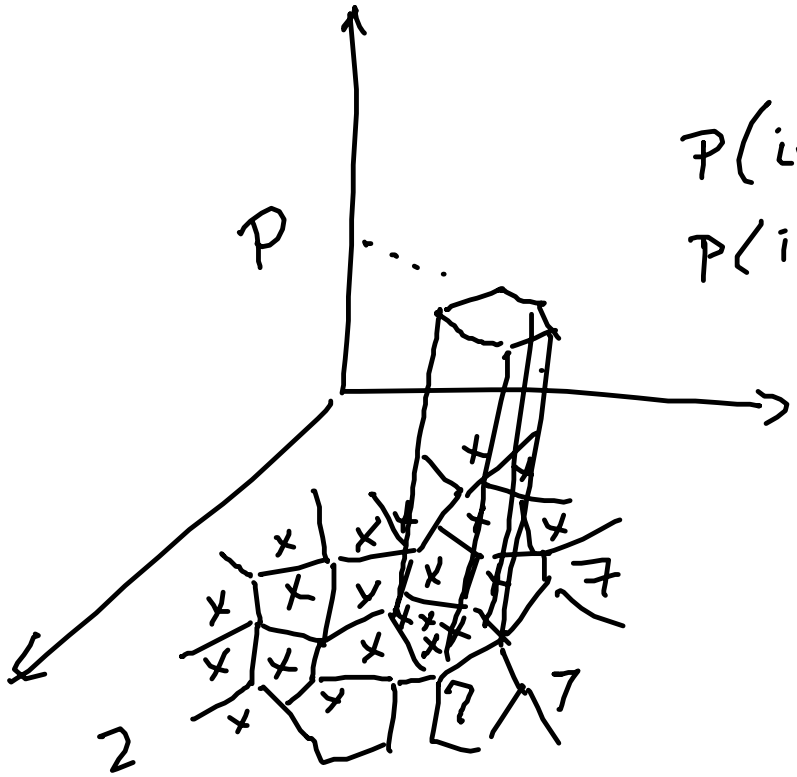
↓

"1" ← Klassifikator


Trainingsmenge
"1"
2
1
"3"
3


1
1
1
1
1
1
1
....

p
Gauss
1
$\mathbb{R}^2$
2



$P(\text{input} \mid 2) = \frac{0,4}{0,2}$

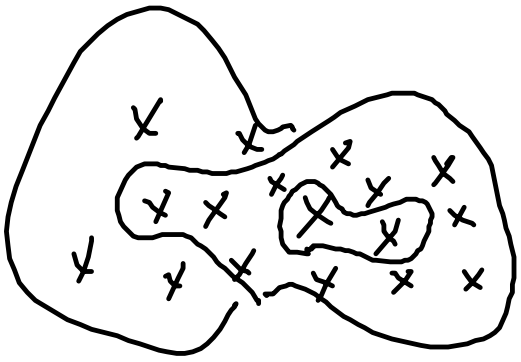
$P(\text{input} \mid 7) = 0.2$

Nearest-Neighbor-Klassifizierung

→ Die Klasse einer neuen Eingabe ist die Klasse des nächsten Nachbarns.

Vorsicht! Ich brauche eine gute Metrik

1 2 3 → 1001001111 / 1011001111



Nachteil: 50,000 Ziffern

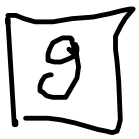
↓

Vorverarbeitung

↓

1000,000 Parameter

Eingabe



50,000 Euklidische Abstände in $\mathbb{R}^{20}$

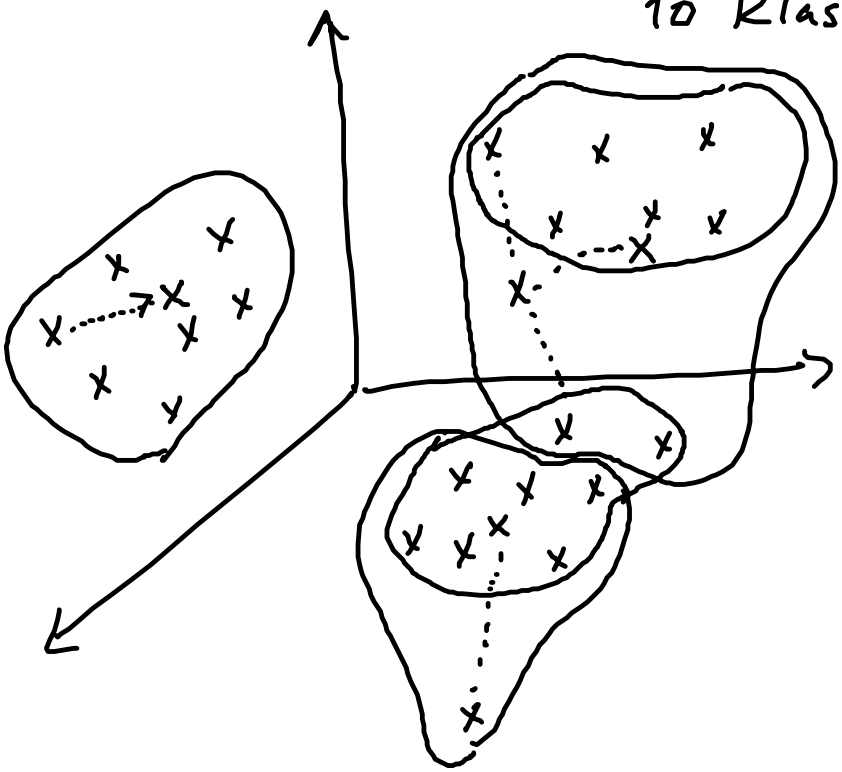
1/10 Sek

Verbesserung: Clustering

LBG - Algorithmus (Expectation Maximization)

K-means

Ungelabelte Daten
10 Klassen

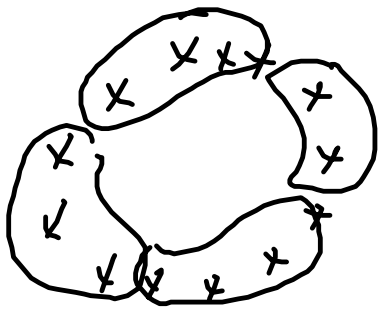


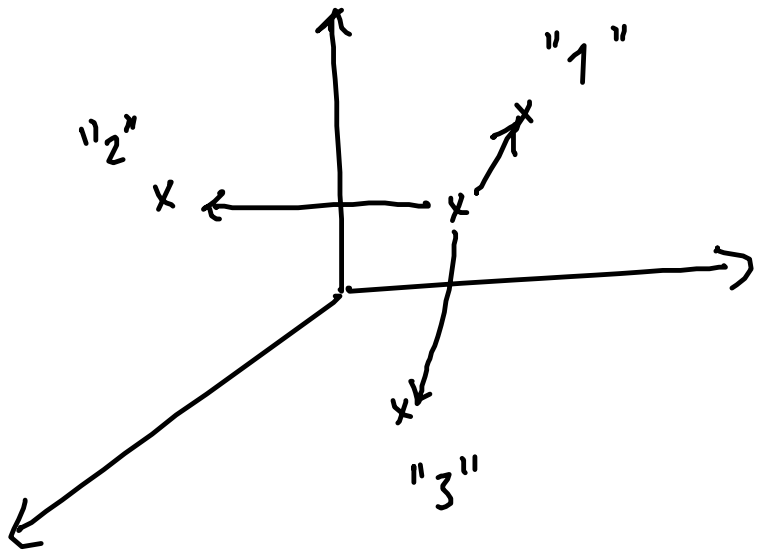
1) – Prototypen werden zufällig verteilt.

Expectation 2) – Die Eingabedaten werden klassifiziert

Maximization 3) – Die Prototypen werden auf den Schwerpunkt der Klassen gelegt.

→ bis keine Änderung mehr enthält
→ bis ich müde bin (es gibt keine Garantie, dass es terminiert).



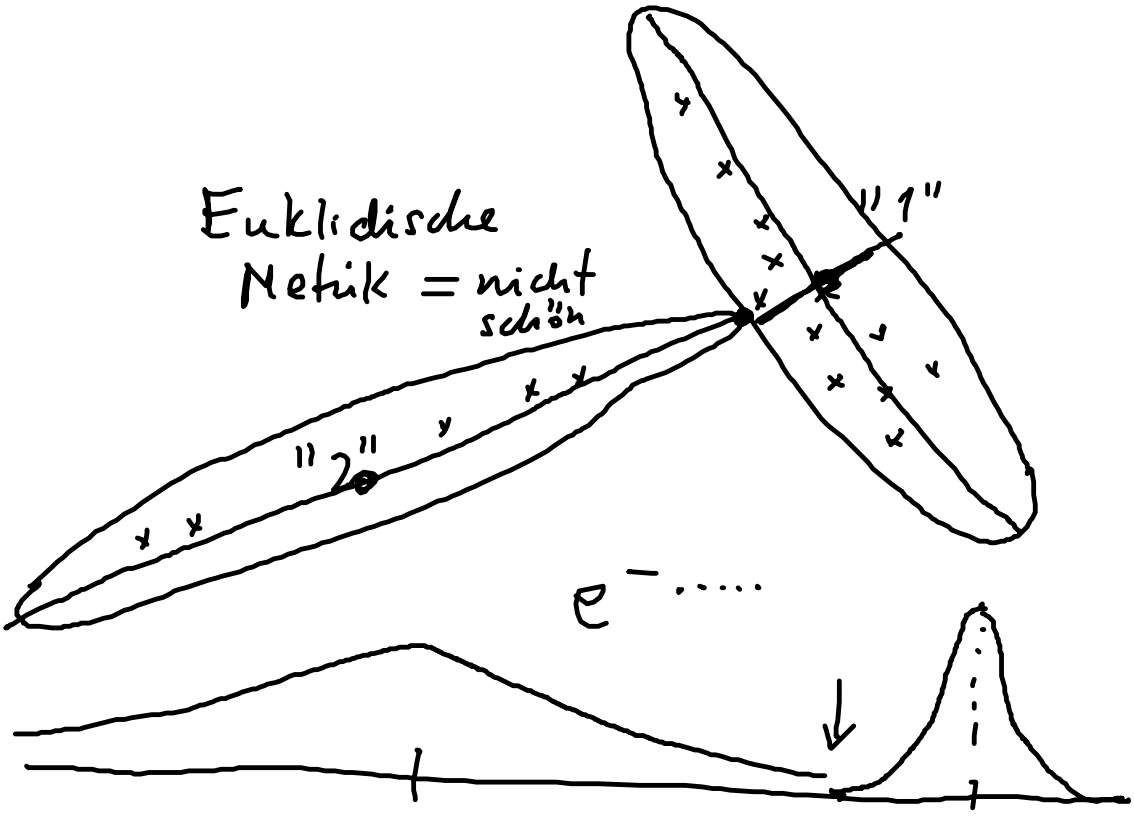


Vorteil

10 Euklidische Abstände

Nachteil:

Ich verliere die Struktur



"1"

K - nearest neighbor

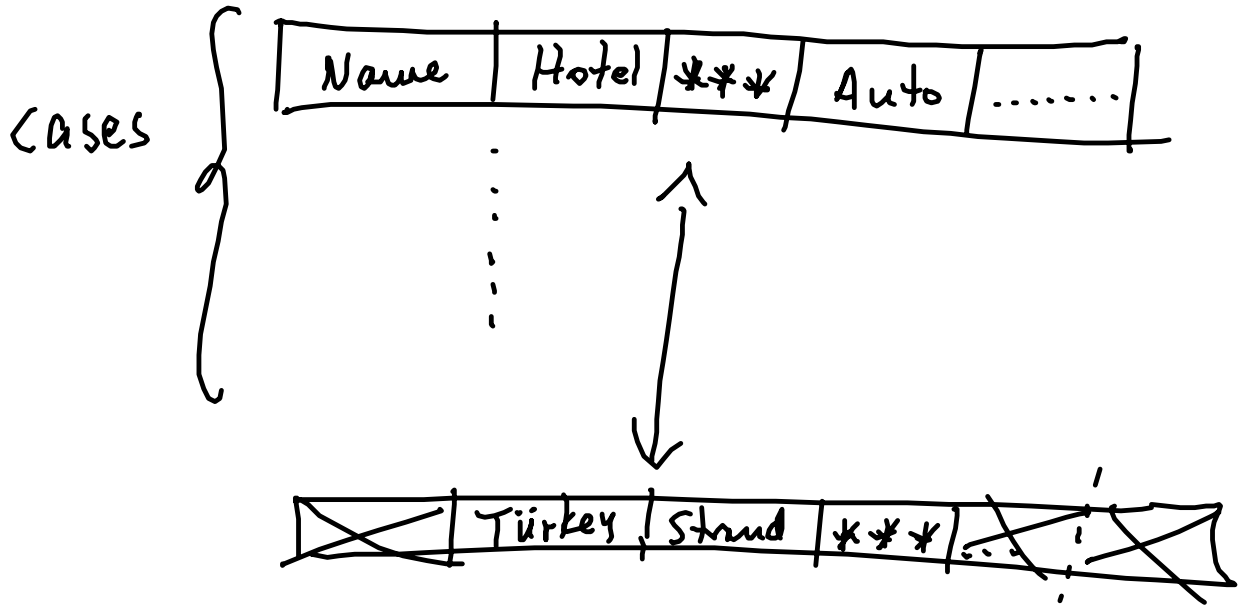
4 - " "

nearest neighbor

"

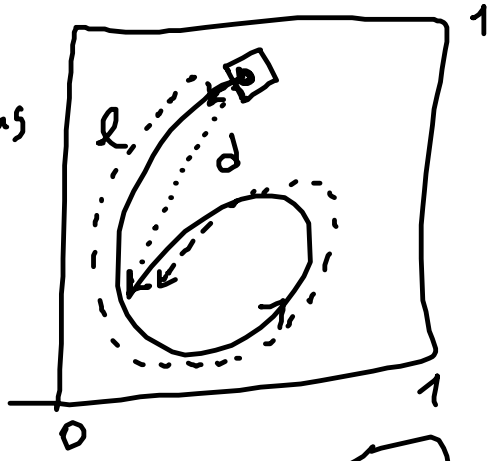
case based reasoning

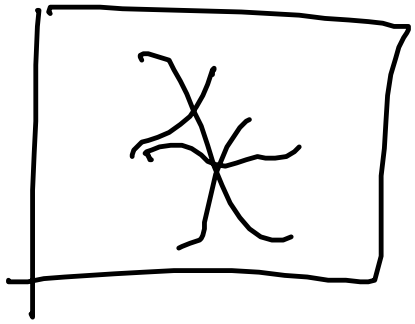
Reisebüro



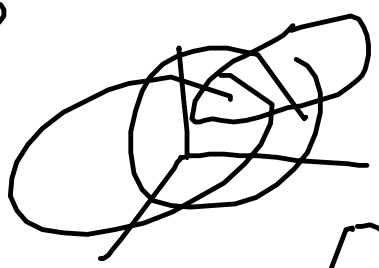
Metrik : Parameter

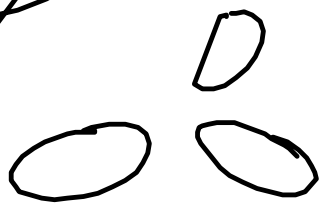
1 Linearitys
360°
l
d/l

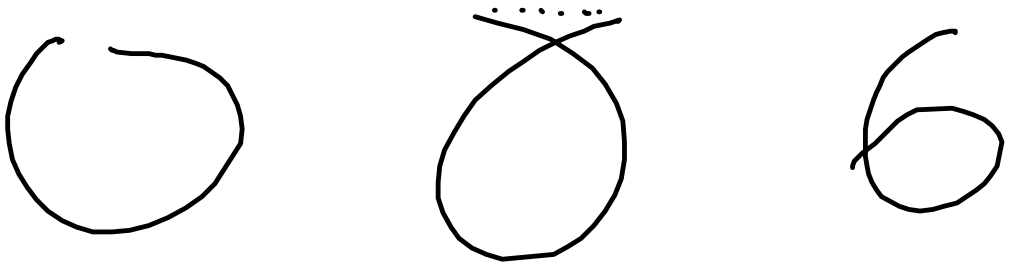


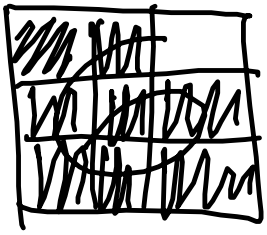


4
Vertikal
horizontal
⋮

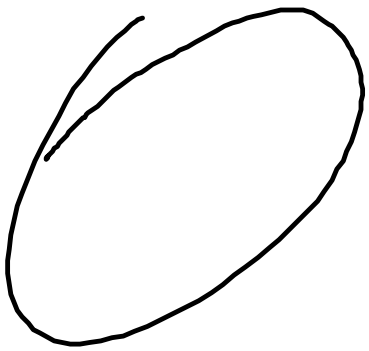








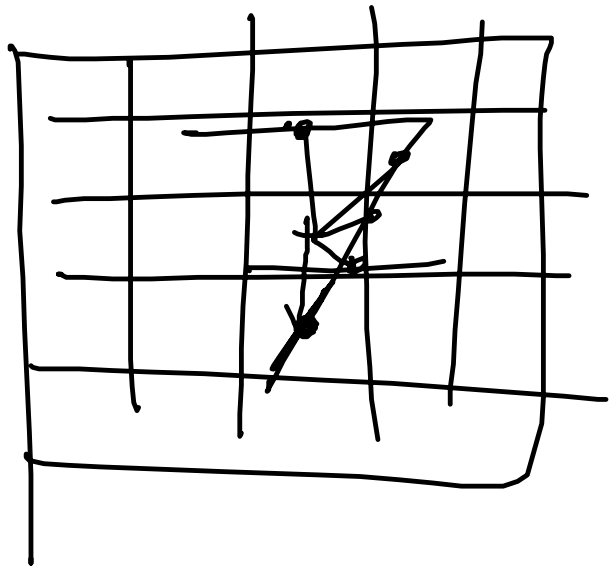
Momente

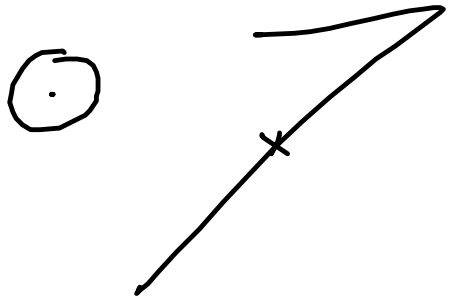


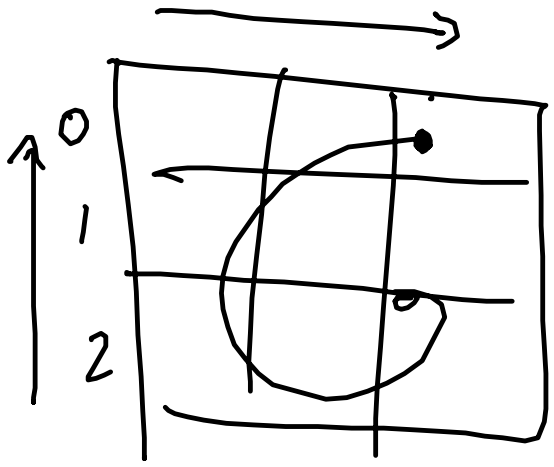
Linearität
Kon. Winkel
L
d/L

$p = 2, 3, 4, \ldots$

$(x_1 - x_0)^p$





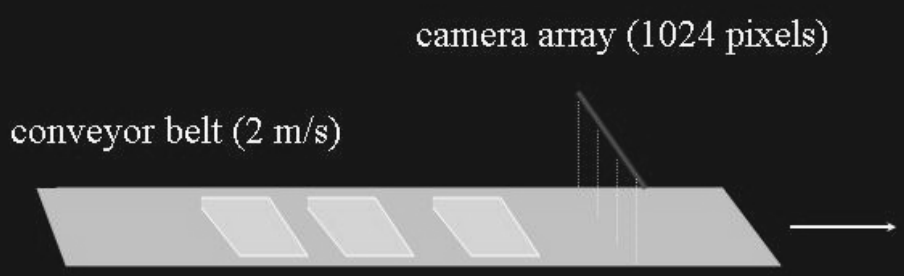


$\mathbb{Q}_2$

But first: Mail sorting machines

- Millions of letters are delivered every day in Germany
- A GFS machine sorts 20,000 flats (large format envelopes) per hour (5-6 pieces per second)
- At least 85% of the pieces should be classified automatically
- The rest semi-automatically
- Miss-classification penalty: 1 day

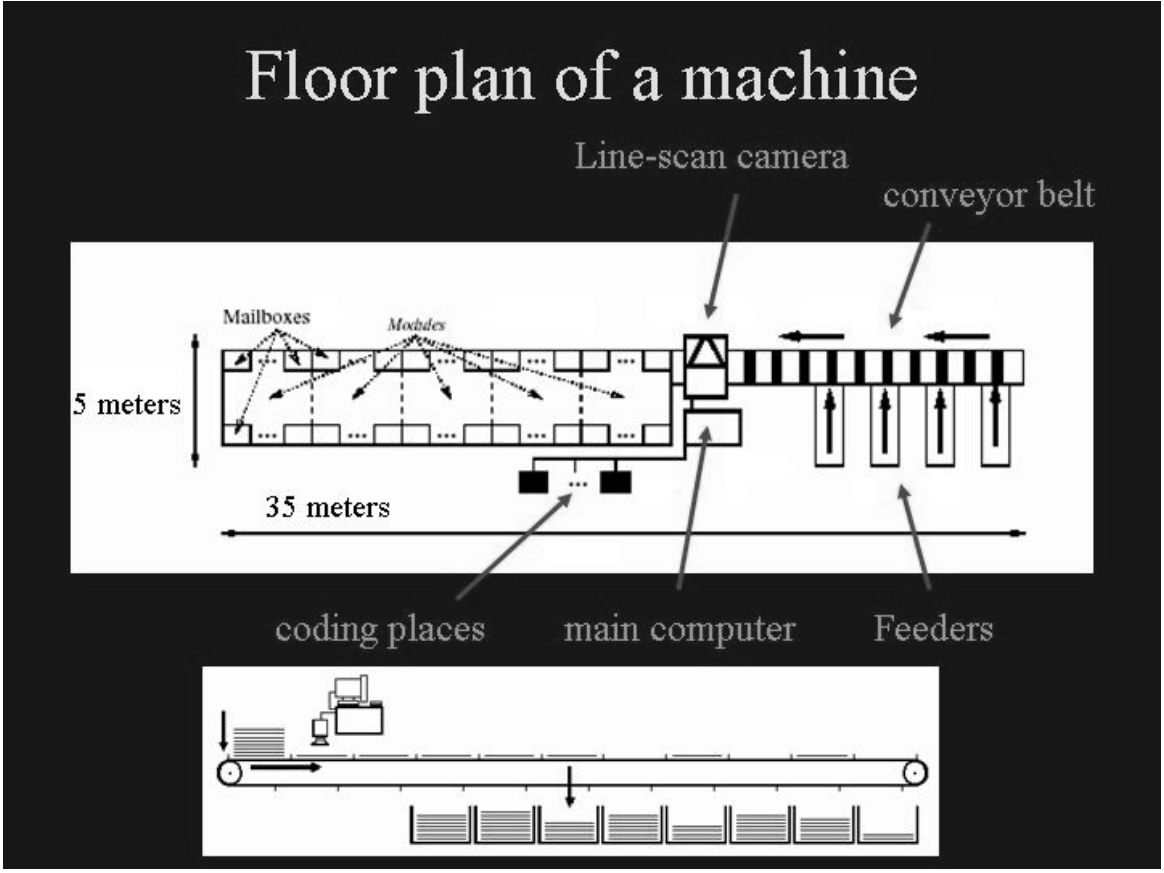
Capturing the data

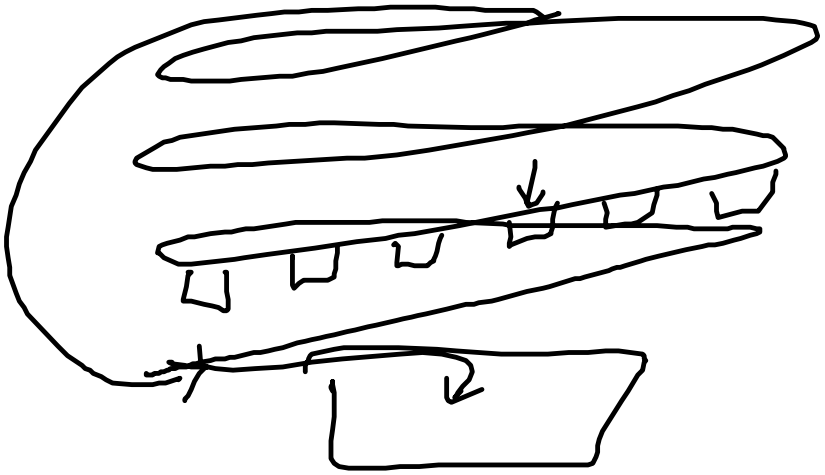


The image is captured by
A computer

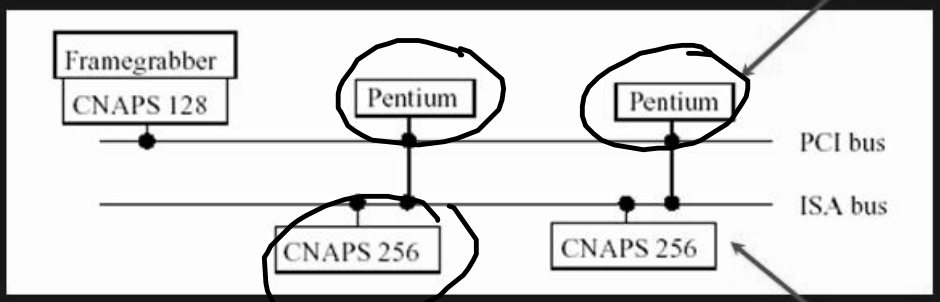
A person can type in the
ZIP code at video coding places

Floor plan of a machine



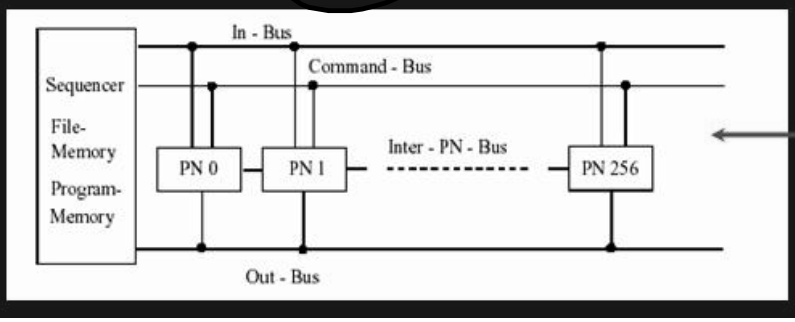


Special hardware



Classifier 1

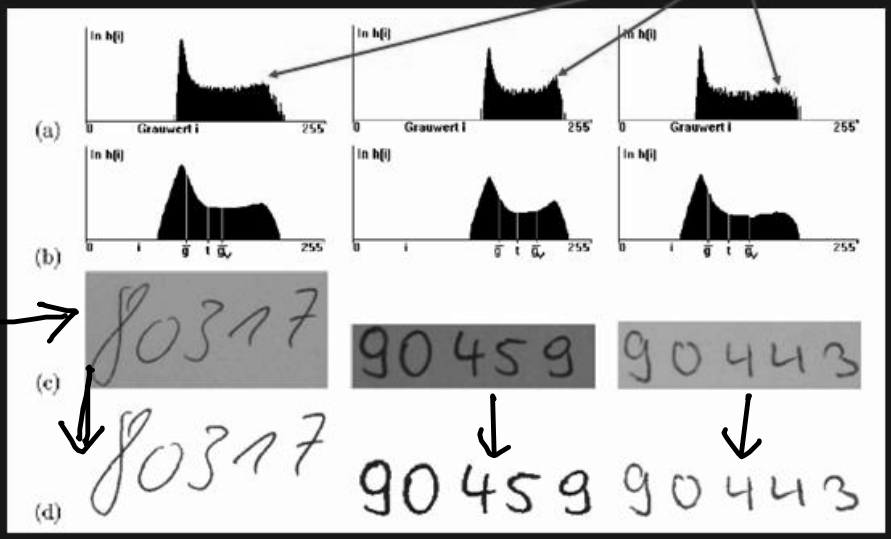
Classifier 2

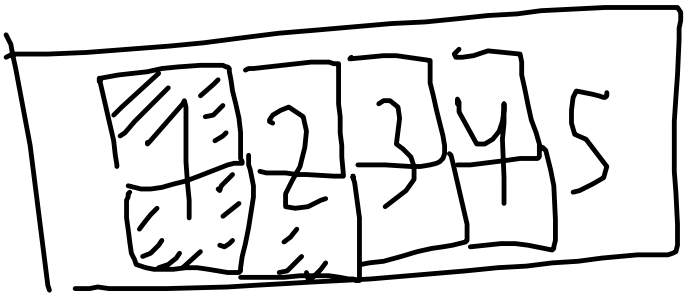


Parallel SIMD machine

Binarization

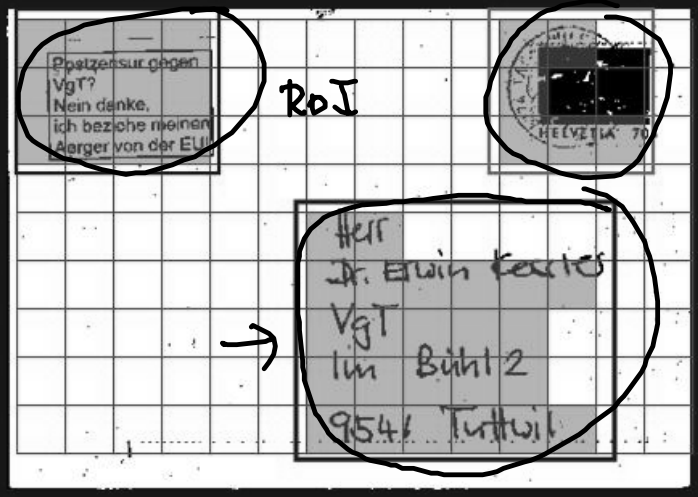
ink





Texture Analysis

Spalten

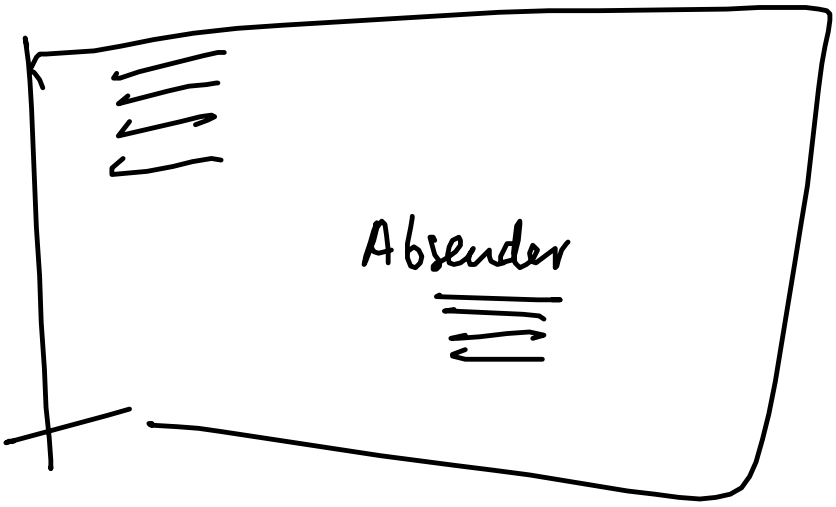


1) Identify regions containing writing

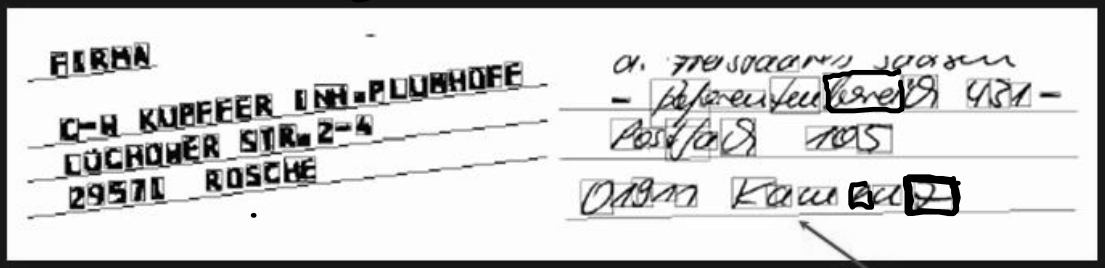
2) Select regions of interest

3) Process according to heuristics

Zeilen



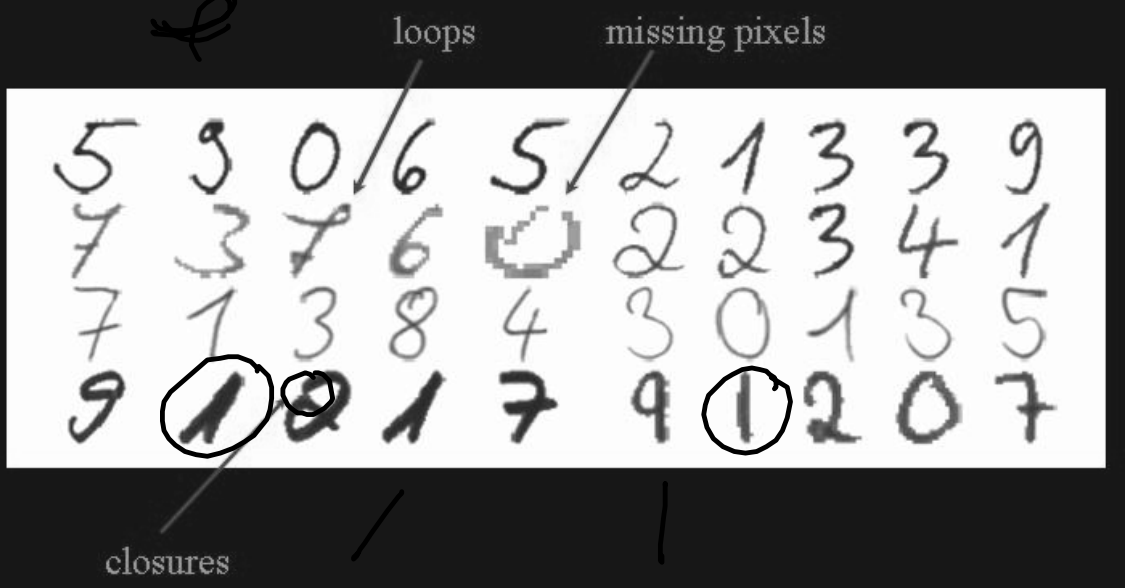
Find lines



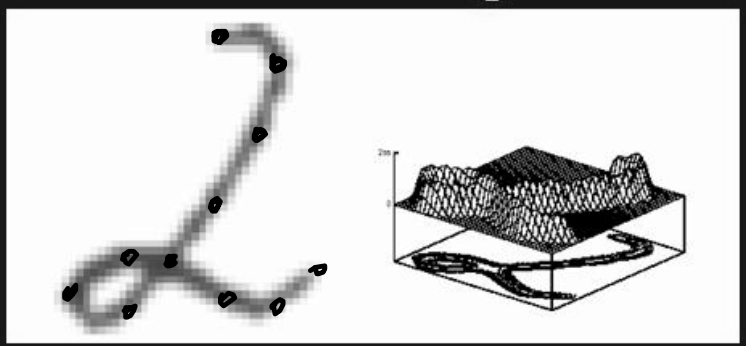
line with the appropriate ZIP features

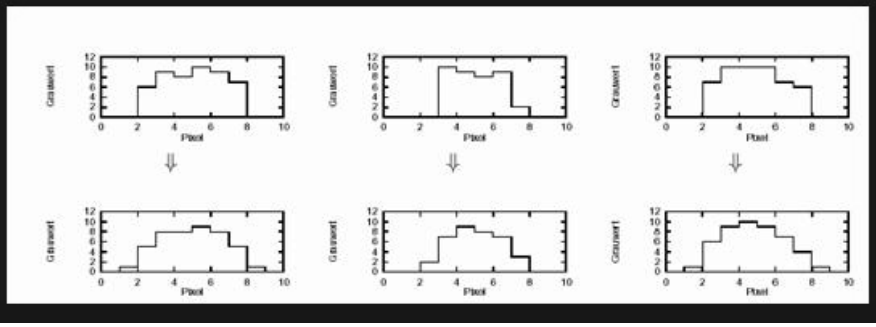
1) Find connected components
2) Bound with boxes
3) Detect orientation of chains of boxes (lines)
4) Distinguish between handwritten and typed text (from the variability and geometry of the boxes)

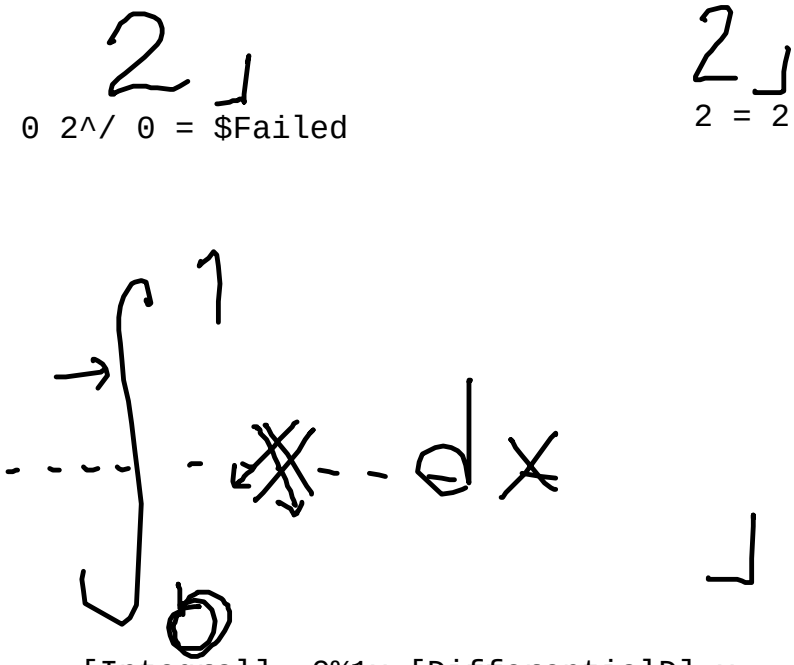
Some typical digit images



First classifier: smooth the gray value image







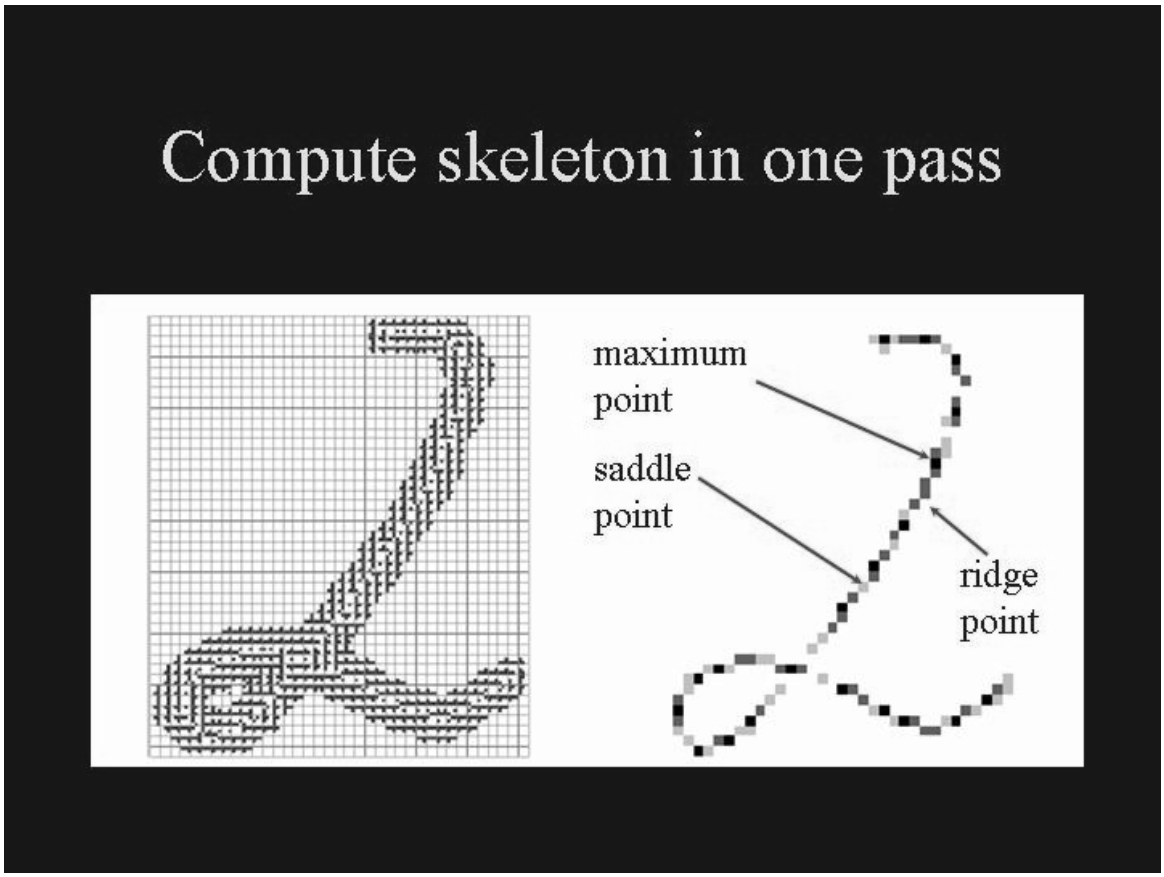
```
0 2^/ 0 = $Failed
```



```
2 = 2
```

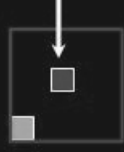


```
[Integral] _0%1x [DifferentialD] x  =
1
-
2
```

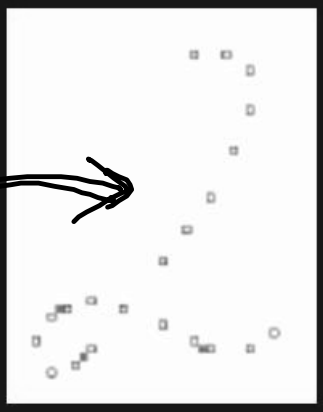


Simplify digit

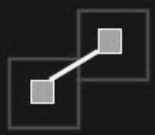
Keep most dominant points

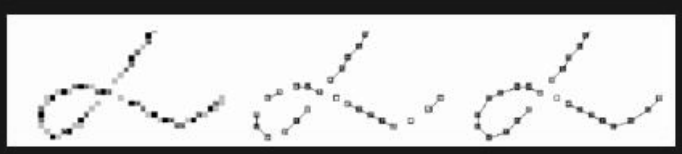


Delete points in the neighborhood



Connecting m×m overlapping neigborhoods



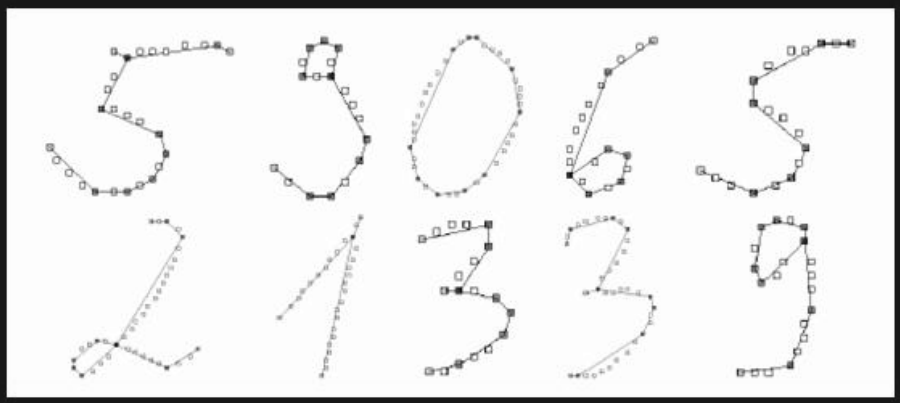


Reconstruct connections (best continuity)

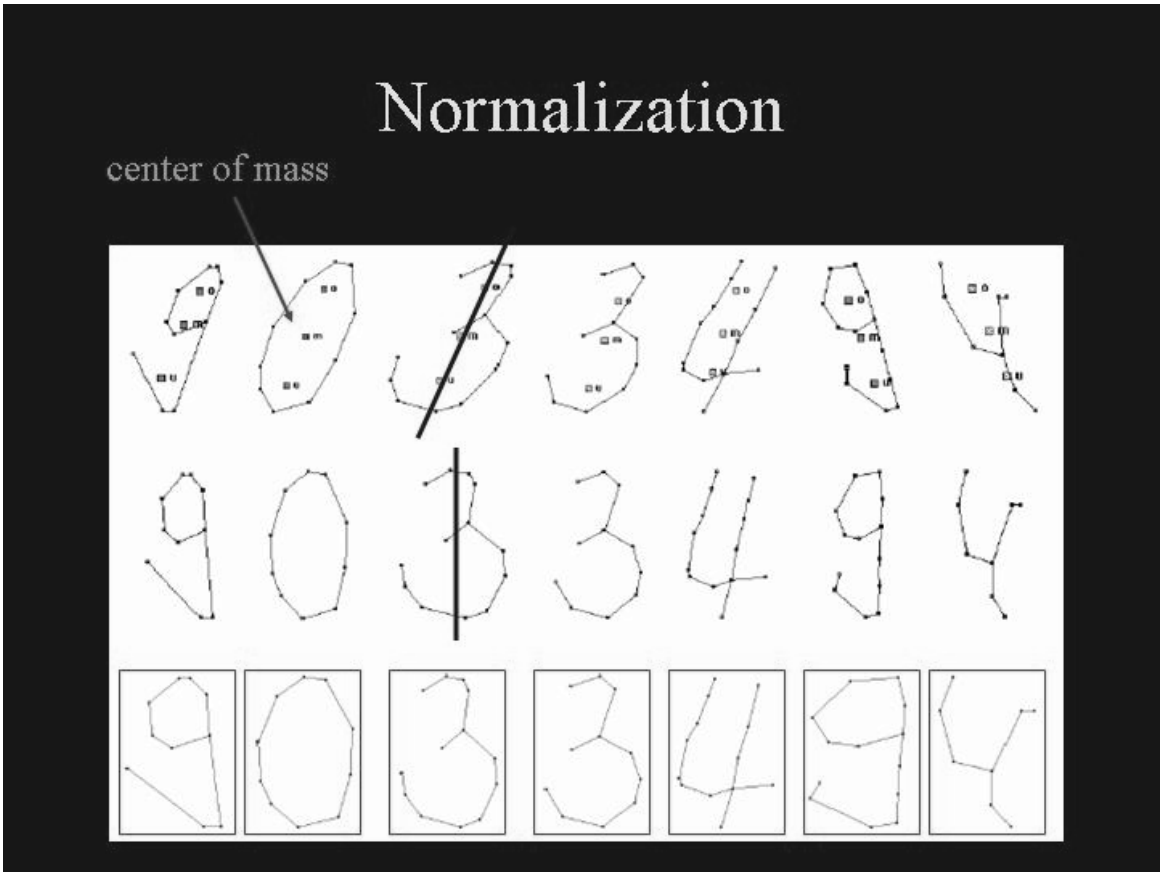
Only a few pixels represent now each digit

Simplify stroke representation

Start new stroke
if covered points are
too far away, or if
angle is to steep



Normalization

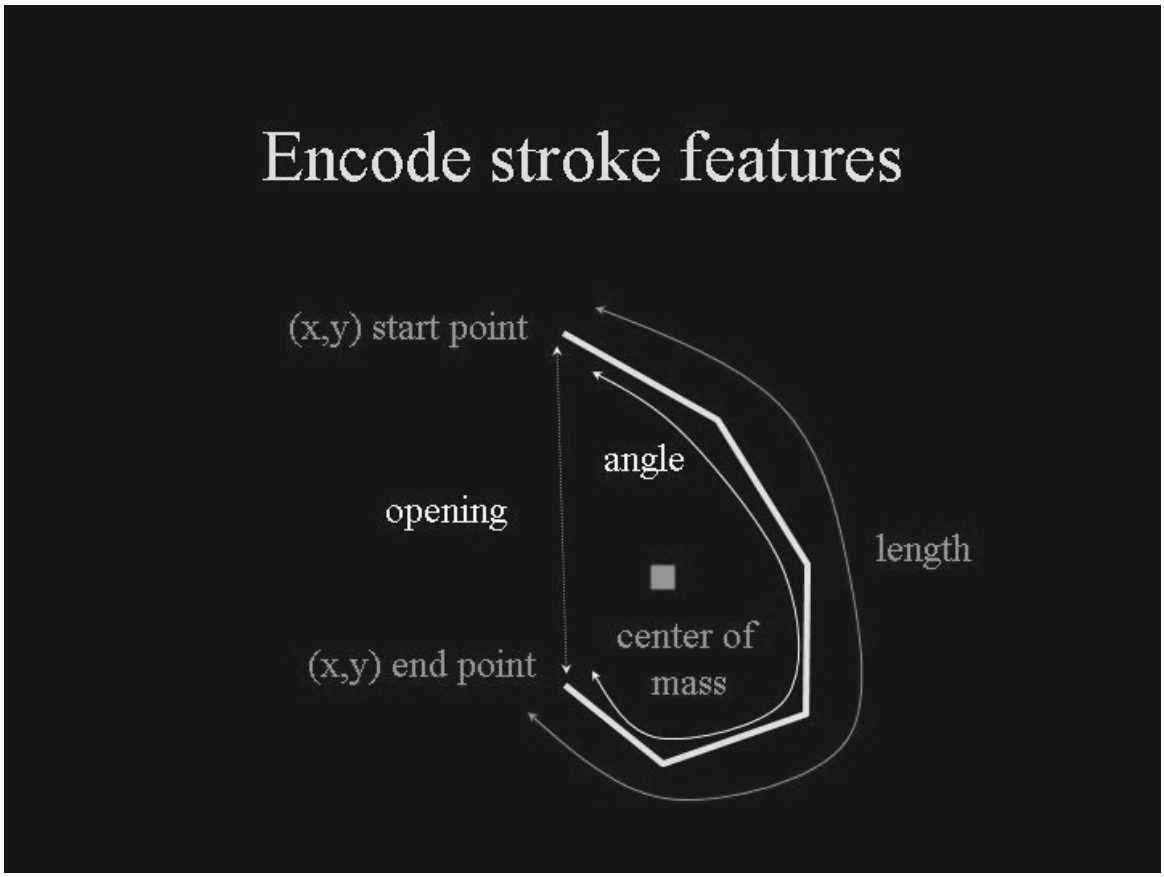


Find strokes

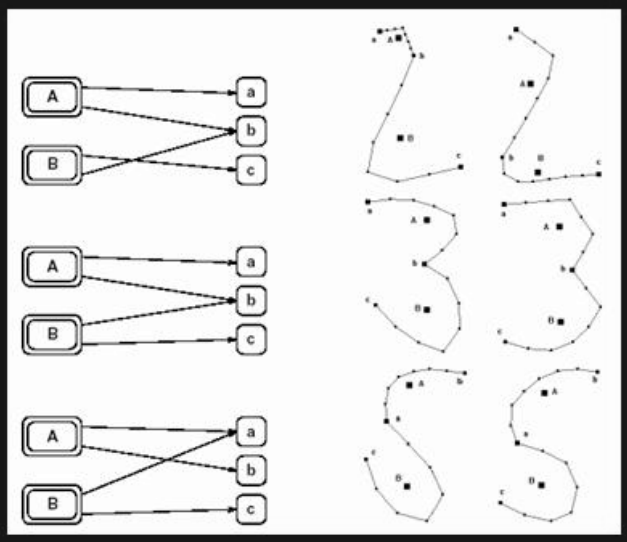
Collect lines if they provide a smooth continuation

Gestalt rules

off-line → on-line



Find model and classify



There is a neural network for each kind of model

Features

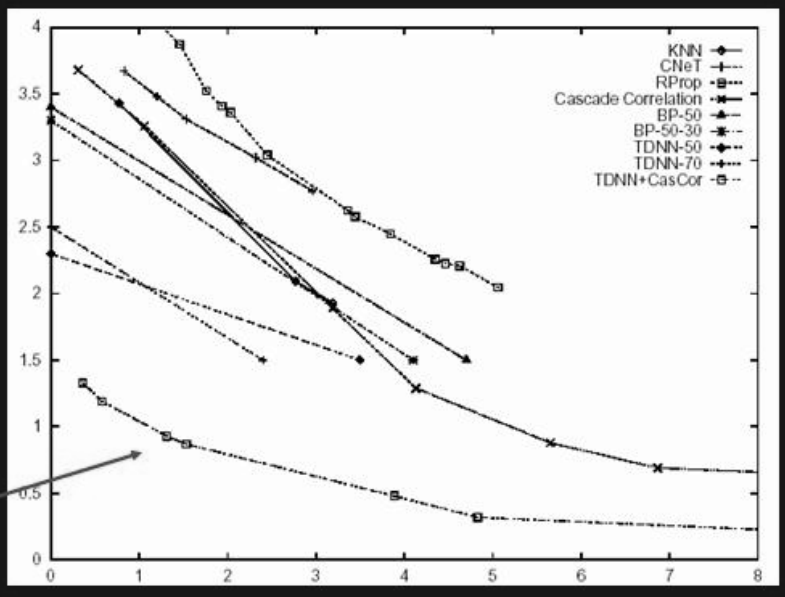
NN

class

Results

substitution rate

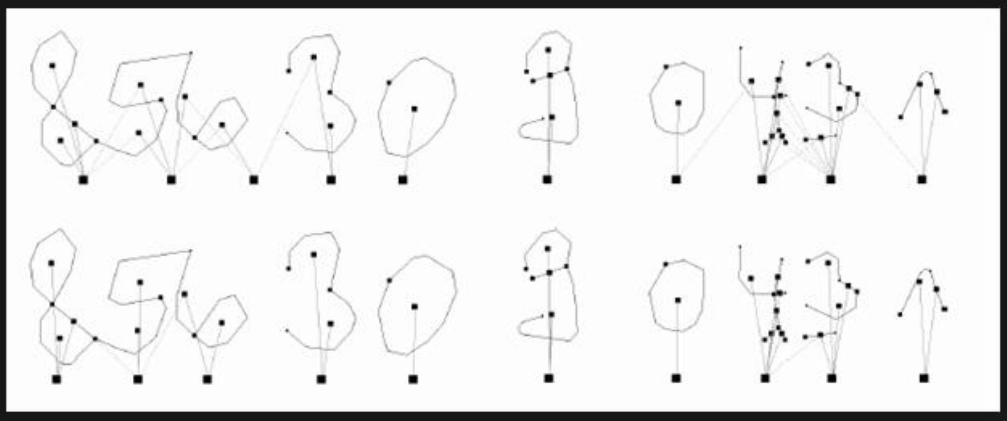
combined classifier



rejection rate

Clustering ZIP strokes

initialization



last iteration